Оптимизация процедуры автоматического пополнения веб-каталога*.

Киселев М. В.
Компания Megaputer Intelligence

В работе изучается влияние применения различных методов классификации текстов на точность автоматического пополнения веб-каталогов [4, 5, 11]. В процессе исследования была выработана стратегия проведения экспериментов, направленных на определение оптимальной классификационной процедуры, а также сформулированы новые методические рекомендации по построению и оценке точности иерархического классификатора. На основе экспериментов с выборкой из содержимого веб-каталога Яндекс была доказана оправданность этих рекомендаций и определены оптимальные с точки зрения точности алгоритмы классификации и формирования векторов признаков, оказавшиеся разными для разных уровней дерева каталога.

The influence of application of different text classification methods on accuracy of automated distribution of internet pages in the thematic web directory nodes [4, 5, 11] has been explored. An experiment strategy aimed at determination of the optimal classification procedure has been designed. New methodological recommendations for creation and accuracy evaluation of hierarchical classifiers have been proposed. On the basis of the experiments performed on a sampled Yandex web directory contents the validity of these recommendations have been proved and the optimal (in terms of accuracy) algorithms for classification and feature vector formation have been determined. It was found that the sets of these algorithms differ for the different web directory levels.

* Данная работа поддерживалась компанией Яндекс (грант №102710).

1. Введение

Экспоненциальный рост объема информации, содержащейся в Интернете и локальных сетях организаций, является причиной все более и более возрастающей трудности поиска необходимых документов и организации их в виде структурированных по смыслу хранилищ. Начиная еще с докомпьютерной эпохи, в качестве эффективного средства смысловой организации массивов документов, обеспечивающего возможность удобного доступа к ним, используются иерархические каталоги. В настоящее время этот подход применяется разнообразными компьютерными системами поддержки поиска и доступа к документам. Вероятно, из всех типов таких систем, лидерами по количеству использующих их людей, являются вебкаталоги, такие как Yandex, Yahoo! или Rambler. В качестве других примеров можно назвать рубрицированные хранилища патентов (например, Всемирной Организации Интеллектуальной Собственности WIPO) или разнообразные компьютерные библиотечные каталоги. Такие системы незаменимы для эффективного поиска и навигации в огромных массивах документов, однако поддержка их полноты, производимая главным образом вручную, становится все более трудоемкой в условиях взрывного роста числа документов, что вызвало интерес к развитию методов автоматического пополнения каталогов. Изучению возможностей повышения эффективности этого процесса и посвящено данное исследование.

Главным инструментом, применяемым для решения этой задачи, являются методы автоматической классификации текстов. В этих методах процедура, относящая новые тексты к тому или иному классу, строится на основе автоматического анализа набора текстов, уже распределенных по классам (называемого обучающим набором). В настоящее время имеется большое разнообразие алгоритмов классификации, основывающихся на самых разных идеях из области математической логики, статистики, искусственного интеллекта и нейронных сетей. Те из них, которые представляются наиболее эффективными с точки зрения классификации текстов, рассмотрены в следующем разделе.

Надо отметить, что задача автоматической классификации текстов является в настоящее время весьма хорошо изученной благодаря усилиям многочисленных исследователей и исследовательских групп. Однако, в абсолютном большинстве этих работ рассматривалась «плоская» одноуровневая классификация, где целевые группы, в которые надо классифицировать документы, не образуют иерархии. Даже в тех работах, где рассматривалось применение автома-

тической классификации к заполнению иерархических каталогов, эта задача, как правило, трактовалась как совокупность подзадач построения бинарных классификаторов, различающих документы, отнесенные к отдельному «листовому» классу дерева каталога от остальных документов. Однако в условиях большого количества документов и узлов дерева каталога этот подход неприменим по многим причинам. В случае «плоской» классификации необходимо решать столько задач классификации, сколько имеется листовых узлов в дереве каталога, и в каждой из этих задач приходится иметь дело со всем объемом обучающих документов, что в случае значительного количества документов приводит к неприемлемо большим временам счета. Если же мы учитываем иерархию, то при классификации документа в некоторый узел дерева используются лишь документы, относящиеся к непосредственно вышестоящему узлу. Кроме того, при плоской классификации распределение обучающих документов оказывается крайне неравномерным – очень небольшой процент относится к выбранному листу, в то время как почти все остальные документы к нему не относятся. В условиях такого перекоса в распределении по целевым классам большинство методов классификации становятся малоэффективными. Также, многие свойства текстов, обеспечивающие возможность точной иерархической классификации, оказываются малоценными в случае плоской классификации. В качестве примера можно привести фрагмент воображаемого каталога с узлами верхнего уровня «Туризм» и «Программное обеспечение» и с их дочерними узлами («Европа», «Африка», «Цены») и («Операционные системы», «Базы данных», «Цены»), соответственно. Наличие слова «цена» в тексте может быть ценным классифицирующим признаком для отнесения к классу «Программное обеспечение/Цены» текстов, уже отнесенных к классу «Программное обеспечение» (например, вследствие наличия в них слова «Windows»), но значительно менее пригодным в случае плоской классификации, так как не даст возможности отличить этот класс от класса «Туризм /Цены».

Одной из целей данной работы было нахождение способов использования специфики, привносимой наличием иерархичности целевых классов, для построения оптимальной процедуры пополнения веб-каталога. Настоящее исследование носило в основном эмпирический характер. Для оценки оптимальности классификаторов использовалось реальное наполнение веб-каталога Яндекс[1], выборка из которого была любезно предоставлена компанией Яндекс специально для целей исследования. Так как иерархическая классификация представляется как суперпозиция задач бинарной классифи-

кации, ключевую роль в настоящей работе играла сравнительная оценка различных бинарных классификаторов в условиях этой задачи, а также настройка их параметров. Разнообразие алгоритмов, предложенных для автоматической классификации текстов (отчасти отраженное в следующем разделе), объясняется в большой степени их неодинаковой эффективностью при разных характеристиках задачи классификации – количестве документов в обучающей выборке, их среднем размере, тематической однородности и т.д. В связи с этим при выборе оптимальной техники для решения какого-то класса задач экспериментальный сравнительный анализ является единственным надежным средством оценки. Насколько мне известно, столь полный, как в данном исследовании, сравнительный анализ разных методов классификации применительно к иерархической классификации веб-страниц еще не проводился. Большую роль в обеспечении возможности такого анализа сыграло появление программных продуктов, объединяющих в рамках одной системы большое количество разнообразных классификационных алгоритмов. Одной из таковых является система автоматического анализа данных PolyAnalyst [10], которая и была использована мной для проведения данного исследования. В рамках этого анализа сравнивались как алгоритмы, часто применяющиеся для классификации текстов, так и наша модификация сравнительно малоизвестного метода, называемого «случайный лес» (random forest). Этот метод был изобретен Л. Брайманом [1] и, насколько мне известно, почти не применялся до сих пор для классификации текстов. В данной работе изучалась эффективность предложенной нами модификации этого алгоритма, показавшей, как будет видно далее, весьма высокие результаты. Наконец, еще один ранее не изучавшийся аспект проблемы автоматического пополнения веб-каталога, которому было уделено внимание в данном исследовании, - это более реалистичная оценка точности тестируемых классификаторов, предполагающая, в частности, что классифицируемые веб-страницы могут относиться как к хостам, содержащим обучающие документы, так и к совершенно новым хостам (что, возможно, является даже более частой ситуацией). Это приводит к нарушению обычного для статистических методов анализа данных предположения, что обучающие и тестовые документы принадлежат одной генеральной совокупности, и накладывает на алгоритмы дополнительные требования устойчивости.

2. Краткий обзор существующих техник классификации текстов.

Процедура автоматической классификации текстов обычно включает две основные части: представление текстов в виде векторов признаков и построение классификатора на созданном массиве векторов. Так как для реализации каждого из этих этапов были предложены разнообразные техники, которые могут использоваться в различных сочетаниях, то и наш обзор естественным образом разбивается на 2 части.

2.1. Методы представления текстов в виде векторов признаков.

Необходимость этого шага (который далее будет называться *векторизацией*) определяется тем обстоятельством, что все методы классификации, которые будут рассмотрены в следующем подразделе, требуют, чтобы классифицируемые объекты были представлены в виде последовательностей чисел одинакового размера и одинакового формата. Эти последовательности и называются векторами признаков.

Описываемые ниже методы могут быть использованы в комбинации друг с другом. Так как влияние их всех на эффективность классификации изучалась в данном исследовании, то чтобы сделать ссылки на них более удобными, присвоим этим методам короткие буквенные идентификаторы, которыми будем помечать их описание.

Во всех рассматриваемых методах каждая позиция в векторе признаков соответствует некоторому объекту, наличествующему (или отсутствующему) в тексте. Разные методы различаются типами объектов, соответствующих позициям в векторе, способами вычисления стоящих в векторе значений и способами уменьшения размеров этих векторов.

Первый выбор, который приходится делать при определении процедуры векторизации, это выбор типов объектов, соответствующих позициям в векторе признаков. Здесь возможны следующие варианты.

WF. Каждая позиция соответствует некоторой форме некоторого слова. Например, словоформам «культура» и «культурами» будут соответствовать разные позиции.

NF. Каждая позиция соответствует нормальной форме некоторого слова. Например, словоформам «музейная» и «музейным»

будет соответствовать одна позиция, а словам «музей» и «музейному» - разные.

SYN. В этом подходе для группирования слов с близкими значениями в рамках одной позиции вектора признаков (и тем самым уменьшения его размерности) используется тезаурус. Для целей данного исследования мной был применен широко известный тезаурус WordNet [6]. Основной структурной единицей этого тезауруса является *синсет* – синонимическая группа слов. Синсеты связаны отношением гипернимии (отношением «общее – частное»). Каждая позиция в векторе соответствует синсету с учетом отношения гипернимии, так что если, например, в тексте присутствует слово «Windows», то это повлияет на позиции вектора, соответствующие синсетам «операционные системы» и «программы».

Кроме того, существуют подходы, в которых позициям в векторе соответствуют устойчивые словосочетания, автоматически находимые в текстах, или даже более сложные сущности, например, названия фирм, имена людей. Однако, эти методы остались за пределами нашего рассмотрения, так как они либо применяются в более узко специализированных задачах, либо требуют столь значительного увеличения времени вычислений, что оказываются малопригодными в случае больших веб-каталогов.

Далее идут методы ограничения размерности векторов признаков, которые могут использоваться в комбинации.

STOP. Использование *стоп-листа*, списка частых неспецифичных слов, не несущих информации о смысле текста, как, например, слова «каждый», «вид», «являться». В вектор признаков не включаются позиции, соответствующие словам из этого списка.

NOUN. Используются только существительные.

IG. Используются лишь те позиции вектора, которые имеют наибольшее значение шенноновской меры взаимной информации с метками целевого класса. Так как эксперименты показали малую ценность этого метода для решения нашей задачи (не было продемонстрировано улучшения точности ни в одном из 10 поставленных предварительных экспериментов) этот метод был исключен мной из рассмотрения. Его подробное описание можно найти в [2].

FREQ. На основе информации об априорных вероятностях встречаемости слов, полученной на большом корпусе текстов, для включения в вектор признаков отбираются только те, чья частота в каком-либо из текстов выходит за пределы (точнее,

превышает) границу доверительного интервала, вычисляемого из предположения о равномерном распределении соответствующего объекта (словоформы, слова или синсета) в текстах.

Наконец, последнее различие между разными процедурами векторизации состоит в способе вычисления величин, стоящих в векторе признаков. Здесь в абсолютном большинстве подходов используются два варианта.

BIN. Вектор содержит 1, если соответствующий объект наличествует в тексте и 0 – если нет.

TFIDF. Применяется более сложная мера, вычисляемая по формуле $tfidf = f \log\left(\frac{D}{D_t}\right)$, где f – частота объекта в тексте, D – общее количество обучающих документов, D_t – количество документов, содержащих данный объект.

Заключительная манипуляция, которая производится всеми классификационными алгоритмами с векторами признаков, - это их нормирование – компоненты всех ненулевых векторов делятся на евклидову длину вектора.

2.2. Методы построения классификаторов на основе совокупности векторов признаков.

Цель этого шага – построение классификатора на основе набора векторов признаков и соответствующих им меток целевых классов. Классификатор – это некоторый алгоритм, принимающий на вход вектор признаков и выдающий метку класса, к которому надо отнести данный вектор, а, следовательно, и транслированный в него текст. Вид этого алгоритма, также как и способ его построения сильно варьируют в разных методах. Для данного исследования были выбраны наиболее известные алгоритмы, обладающие наибольшим диапазоном применимости, и, вместе с тем, достаточно быстрые, чтобы быть пригодными для анализа больших веб-каталогов. Мы также будем помечать их короткими буквенными идентификаторами.

CLLR. Метод классификации на основе линейной регрессии. Данный метод применим в основном для бинарной классификации – т.е., когда имеется только два целевых класса, помечаемых числами 0 и 1. Задача классификации рассматривается как задача построения линейной регрессии, где зависимой переменной является этот набор нулей и единиц, а независимыми переменными - компоненты обучающих векторов признаков. При этом производится пошаговый отбор независимых переменных, включаемых в регрессионную модель, на основе значения F-статистики их регрессионных

коэффициентов. После построения регрессионной модели выбирается величина порога возвращаемого ей значения, так что если это значение выше порога, соответствующий вектор относится к классу 1, иначе – к классу 0. Порог выбирается на основе критерия минимизации ошибки классификации, возможно, взвешенной, если цена неправильной классификации векторов, относящихся к разным классам, различна. Это упрощенный вариант метода логистической регрессии (описание см., например, в [8]), обычно несколько менее точный, но на порядки быстрее работающий, так как в случае логистической регрессии строимая регрессионная модель не линейна по отношению к регрессионным коэффициентам.

DT. Деревья решений (decision trees). Вероятно, самый популярный метод из инструментария data mining. Строится древовидный классификатор, при применении которого вектора признаков перемещаются от корня к листьям. В каждом узле этого дерева стоит некоторый критерий на один из компонент вектора. В зависимости от выполнения или невыполнения этого критерия вектор продвигается к тому или иному нижележащему узлу, пока не достигнет одного из листовых узлов. Значение, помечающее этот листовой узел, и становится предсказанной меткой класса для данного вектора. Отбор очередного компонента вектора, с которым будет связан критерий, стоящий в данном узле, может производиться на основе различных методов. Мной использовался метод максимизации взаимной информации [2] компоненты вектора и метки целевого класса. Также, вместо обычно применяемой процедуры усечения полного дерева решений использовался статистический хи-квадрат критерий расщепления, позволяющий избегать статистически незначимых расщеплений и чрезмерного роста дерева.

NN. Алгоритм «ближайших соседей» (nearest neighbors). Этот классификатор делает прогноз на основе N обучающих векторов, ближайших к классифицируемому вектору. В качестве прогнозируемого класса берется класс, наиболее частый среди этих N векторов, либо класс, имеющий наибольший вес, если соседние вектора взвешиваются величиной, обратно пропорциональной расстоянию до классифицируемого вектора. В качестве расстояния используется евклидово расстояние в некотором подпространстве векторов признаков. Набор измерений, составляющих это подпространство, параметр N, а также некоторые другие параметры алгоритма часто подбираются некоторой оптимизационной процедурой, минимизирующей общую ошибку предсказания на обучающем наборе. Мной был использован для этого генетический алгоритм [7].

NB. «Наивный байесовский» (Naïve Bayes) классификатор [9]. Классифицирует на основе определения условных вероятностей отнесения вектора к тому или иному классу при определенных условиях на значения компонент вектора признаков.

SVM. Классификатор, использующий *механизм поддерживающих векторов* (support vector machines) [3]. Этот метод основан на нахождении в пространстве признаков гиперплоскости, которая разделяла бы точки (=вектора), относящиеся к разным классам, при чем так, чтобы минимальное расстояние до нее какой-либо из этих точек было максимальным. Если говорить точнее, мной применялся вариант SVM, использующий скалярное умножение векторов в качестве функции свертки.

RF. Классификатор «случайный лес» (random forest) [1]. В этом методе классификатор строится в виде большого количества деревьев решений, причем в качестве предсказанного для данного вектора класса выбирается тот, который предсказывается большинством из этих деревьев. Деревья, составляющие классификатор, строятся независимо, так, что при расщеплении всех узлов всех деревьев используется лишь небольшое и каждый раз случайно выбираемое подмножество признаков, составляющих вектора. В результате того, что прогноз делается достаточно большим ансамблем независимых классификаторов, достигается статистическая надежность прогноза даже в случае, когда размерность векторов признаков превышает количество обучающих текстов. В данном исследовании была применена наша модификация этого метода, заключающаяся в том, что вместо критерия Джини (Gini index) для расщепления использовался критерий взаимной информации, что, как показали предварительные эксперименты, приводит к некоторому увеличению точности классификации в случае классификации текстов.

Перед тем, как перейти к изложению идей и методики исследования, в заключение упомяну три наиболее близкие к его теме работы, посвященные применению описанных выше методов к иерархической классификации текстов. В работе [5] приводятся результаты одной из первых попыток применения метода SVM к автоматическому пополнению веб-каталога. Исследование [11] посвящено изучению вопроса, как знание о том, каким образом тексты организованы в одном иерархическом хранилище (например, в иерархии «домен – хост – директории – файлы»), может помочь их автоматическому размещению в другой классификационной иерархии (например, в веб-каталоге). Наконец, в [4] обсуждается метод понижения размерности пространства признаков на основе выделения кла-

стеров признаков, статистически значимо связанных между собой в обучающих текстах.

3. Методика выбора оптимального классификатора для автоматического пополнения веб-каталога.

В предыдущем разделе были перечислены различные методики построения классификаторов, потенциально применимых для автоматического пополнения веб-каталогов. Поскольку они могут быть использованы в различных комбинациях, нахождение оптимального сочетания является непростой задачей. Для того, чтобы убрать из рассмотрения наименее перспективные методы, было проведено предварительное исследование влияния использования каждого из методов при случайном выборе остальных методов и узла каталога, относительно которого производится классификация. Для каждого метода было поставлено 6 таких экспериментов. Если во всех экспериментах применение метода ухудшало точность классификации, данный метод устранялся из рассмотрения. При априорной равновероятности исходов «ухудшит – не ухудшит» это соответствует уровню достоверности $1/2^6 = 0.0156$. Из всех перечисленных в разделе 2 методов этот тест не прошел лишь **IG**, – ограничение количества признаков по критерию взаимной информации всегда ухудшало результат даже в случае небольших обучающих наборов, где, как ожидалось, его применение могло бы помочь бороться с избыточными степенями свободы классификатора. Вероятно, это можно объяснить тем, что большинство методов построения классификаторов (кроме **NB** и **NN**) имеет достаточно надежные встроенные средства обеспечения статистической значимости классификатора.

Итак, после исключения метода **IG** наше пространство оптимизации выглядит как 6-мерный дискретный куб с размерностями, соответствующими взаимоисключающим методам:

1. **WF/NF/SYN**
2. **STOP/-**
3. **NOUN/-**
4. **FREQ/-**
5. **BIN/TFIDF**
6. **CLLR/DT/NN/NB/SVM/RF**.

Здесь прочерк означает неприменение данного метода.

Тем самым, общее количество возможных комбинаций составляет 3*2*2*2*2*6 = 288. Так как каждый из этих 288 экспериментов должен быть проведен для нескольких различных узлов дерева каталога, получается, что общее количество экспериментов для реализации полной схемы оптимизации должно составлять несколько тысяч. Еще на этапе подготовки исследования стало ясно, что такая стратегия не может быть использована по соображениям нехватки времени. По этой причине была применена другая схема, основывающаяся на предположении о независимости влияния выбора метода, соответствующего каждому измерению пространства оптимизации, на результат. А именно, эксперименты организовывались в серии, имеющие в пространстве оптимизации вид «звезд». Сначала случайным образом выбиралась базовая точка – центр звезды, а потом делались эксперименты, соответствующие всем точкам пространства оптимизации, у которых только одна координата отличается от координат базовой точки. При этом каждая такая серия для одного узла дерева каталога состоит из 1 базового эксперимента и 2+1+1+1+1+5=11 экспериментах на лучах звезды, т.е. всего 12 экспериментов. Такая постановка экспериментов позволяет выделить влияние каждого отдельного метода с целью определения методов, дающих в нашей задаче значимое улучшение.

Так как одной из целей данной работы было понимание того, каким образом иерархическая структура каталога может быть использована для повышения точности классификации, мной проверялась следующая рабочая гипотеза. Основываясь на том факте, что в условиях разных параметров задачи классификации (размер и количество обучающих текстов, их тематическая однородность и пр.) наиболее оптимальными являются разные методы классификации и учитывая, что задачи классификации текстов в узлы каталога, лежащие около корня, и в листовые узлы дерева, очень сильно разнятся по этим параметрам, я ожидал, что на разных уровнях каталога выбор оптимальной классифицирующей процедуры должен быть различен. Так как большая часть исследуемого веб-каталога является трехуровневой, для проверки этой гипотезы одинаковые серии экспериментов по схеме, описанной выше, ставились для узлов первого, второго и третьего (а также более глубоких, если таковые имелись) уровней. Насколько известно автору, подобные исследования до сих пор не проводились – исследовалась эффективность одного классификатора на всех иерархических уровнях.

Наконец, использованная мной методика оценки точности конкретного выбранного метода классификации, также содержала один практически важный элемент, по-видимому, ускользавший до сих

пор от внимания исследователей. Дело в том, что обычные способы оценки точности классификатора, основанные на случайном разделении массива имеющихся документов с известным распределением по классам на обучающую и тестовую выборки, не совсем соответствуют реалиям процесса автоматического пополнения веб-каталога. Связано это с тем, что, как подсказывает здравый смысл, документы, принадлежащие одному хосту, должны быть в гораздо большей степени однородны, чем документы, принадлежащие разным хостам, даже если они и отнесены к одному узлу каталога. Если это действительно так, то классификатор должен оцениваться по отношению к двум разным типам ошибок: ошибки классификации документов с тех же хостов, что и обучающие документы (мы будем называть их K-ошибками) и ошибки при классификации страниц с совсем новых хостов, документы с которых не использовались при построении классификатора (U-ошибки). При этом совсем не обязательно классификатор с наименьшей ошибкой первого типа окажется наиболее точным и в терминах ошибки второго типа. Это можно проиллюстрировать примером из области построения числовых регрессионных моделей, если сравнить линейную регрессионную модель и полиномиальную модель достаточно высокой степени. Полиномиальная модель за счет больших степеней свободы будет точнее описывать данные, давая более точные предсказания значения зависимой переменной для тестовых данных, лежащих в той же области значений, что и обучающие данные (разумеется, если они относятся к той же генеральной совокупности). При этом, если мы будем тестировать ее на данных, лежащих на некотором расстоянии от области значений обучающих данных, то просто в силу наличия в регрессионной модели высоких степеней независимых переменных такая модель будет давать бессмысленные результаты, очень сильно отличающиеся от средних значений зависимой переменной в обучающих данных, в то время как линейная модель в этом случае окажется более точной. Таким образом, задача автоматического пополнения веб-каталогов должна накладывать на используемые классификаторы специфическое требование устойчивости классификации текстов, отличных по своим статистическим свойствам от текстов, использованных при построении классификатора. Чтобы проверить оправданность этого предположения, я в каждом эксперименте разбивал документы на 3 группы - обучающие документы, тестовые документы с тех же хостов, что и обучающие документы, и тестовые документы с новых хостов. При этом точность классификатора измерялась по отдельности на обеих группах тестовых документов. Осталось сказать, какая величина использовалась для оценки точно-

сти. Вообще говоря, в разных приложениях используются разные оценки. Например, если речь идет о бинарной классификации на 2 сильно неравные группы текстов, как в нашем случае, то часто в качестве мер точности используются доля правильно классифицированных документов среди документов, классифицированных в меньшую группу, и доля правильно классифицированных документов среди документов из меньшей группы. Это удобные показатели, имеющие ясный смысл. Их неудобство в нашем случае заключается в том, что они применяются именно в сочетании, а нам для целей оптимизации необходим единственный параметр. В качестве такого единственного параметра часто используется доля правильно классифицированных текстов, но значения этого параметра в случае двух классов, сильно неравных по числу документов, являются мало показательными. Например, если один класс в девять раз больше другого, то 90% правильно классифицированных текстов кажутся хорошим результатом, хотя такой показатель достигается, если мы просто отнесем все документы к большему классу. Поэтому я в данной работе использовал показатель, называемый *эффективностью классификации*, который определяется как $\frac{N_{corr} - N_{\max}}{N_{\min}}$, где N_{corr} – число правильно классифицированных документов, N_{max} – число документов в большем классе, N_{min} – число документов в меньшем классе. Легко видеть, что это нормированный параметр, не зависящий от соотношения числа документов в целевых классах. Если мы отнесем все документы к большему классу, то получим эффективность классификации, равную 0, если правильно классифицированы будут все документы, она станет равна 1. Удобно выражать эту величину в процентах.

Тем самым, я изложил использованную в данной работе методологию. В следующем разделе будет описано, что дала проверка с ее помощью изложенных выше гипотез, произведенная на реальном наполнении веб-каталога Яндекс, и к каким конкретным рекомендациям по выбору оптимальной стратегии автоматического пополнения каталога она привела.

4. Эмпирическое сравнение классификаторов на содержимом веб-каталога Яндекс.

Для реализации идей, изложенных в предыдущем разделе, компанией Яндекс была предоставлена выборка из ее веб-каталога, содержащая 1457629 страниц с 45796 хостов, общим объемом 40GB. Од-

нако, по техническим причинам были использованы не все эти файлы, что определялось в первую очередь функциональными ограничениями программной платформы, выбранной для этого исследования – системой анализа данных и текстов PolyAnalyst производства компании Megaputer Intelligence. Во-первых, в этой системе нет средств работы с файлами в формате PDF, а во-вторых, не все из выбранных для тестирования алгоритмов и методов создания векторов признаков реализованы в системе как мультиязычные, будучи ограничены английским языком. Не желая приносить в жертву полноту исследования и учитывая тот факт, что англоязычных текстов в выборке также было очень много (около 15%), мной было принято решение пойти на эти ограничения. Таким образом, после отбрасывания PDF файлов, не англоязычных файлов, а также файлов с незначительным текстовым содержимым (<100 байт) выборка стала составлять 49734 текста с 8932 хостов.

Для каждого из этих файлов было известно их положение в дереве веб-каталога. Распределение файлов по дереву каталога оказалось неоднозначным в том смысле, что один файл мог относиться к разным узлам каталога, пути от которых к корню дерева пересекались только в корне. Считается, что файл, принадлежащий некоторому узлу, принадлежит также всем узлам, лежащим на пути от этого узла к корню. Большая часть путей от корня к листьям в изучаемом каталоге состоит из 3 узлов (не считая корня), так что каталог можно приблизительно считать трехуровневым. Узлы, помеченные как «Прочее» или «Универсальное» были из каталога удалены, а входящие в них тексты перенесены в их родительский узел. Также были исключены из рассмотрения узлы верхнего уровня «СМИ» и «Справки» как тематически неспецифичные. После этого в каталоге осталось 8 узлов верхнего уровня: «Культура», «Учеба», «Общество», «Отдых», «Hi-Tech», «Бизнес», «Развлечения» и «Дом». По этой причине было решено провести по 8 одинаковых серий экспериментов на каждом из 3 уровней каталога. Как было сказано в предыдущем разделе, количество экспериментов в серии равно 12, так что общее количество экспериментов в данном исследовании составило 12*8*3=288. На уровнях 2 и 3 каталога узлы для экспериментов были отобраны случайным образом (точнее говоря, наряду с узлами 3 уровня в процесс случайного отбора были включены и узлы более глубоких уровней). Такими узлами оказались

на втором уровне:

- Hardware (родитель – Hi-Tech)
- Мобильная связь (родитель – Hi-Tech)
- Спорт (родитель – Развлечения)

Таблица 1. Базовые точки серий экспериментов (обозначения см. в Разделе 2).

Типы объектов	**STOP**	**FREQ**	**TFIDF**	**NOUN**	алгоритм
WF	нет	Да	Нет	Да	**NB**
NF	да	Да	Нет	нет	**RF**
SYN	да	нет	Да	Да	**DT**
SYN	да	Да	Да	Да	**DT**
SYN	нет	нет	Нет	Да	**RF**
WF	нет	нет	Нет	Да	**CLLR**
WF	нет	нет	Да	нет	**DT**
NF	нет	нет	Да	нет	**SVM**

- Фотография (родитель – Культура)
- Игры (родитель – Развлечения)
- Власть (родитель – Общество)
- Покупки (родитель – Дом)
- Где развлечься(родитель – Отдых)

на третьем уровне:

- Аквариум (родитель – Домашние животные)
- Запчасти (родитель – Авто)
- Каталоги (родитель – Интернет)
- Философия (родитель – Гуманитарные науки)
- Иммиграционные услуги(родитель – Эмиграция/Иммиграция)
- Рассказы (родитель – Юмор)
- Автоспорт (родитель – Спорт)
- Университеты (родитель – Высшее образование)

Центральные точки серий экспериментов в оптимизационном пространстве также выбирались случайно (см. Таблицу 1.).

Для каждой серии экспериментов на каждом уровне отбирались документы, принадлежащие родителю выбранного узла (или все документы - для экспериментов в узлах первого уровня). Составлялся список хостов, которым принадлежат эти документы. Среди них отбиралась треть хостов, документы с которых использовались для определения U-ошибок классификаторов. Среди документов оставшихся хостов отбиралась треть для оценки K-ошибок. Остальные документы использовались как обучающие. Ошибка, показываемая классификаторами на обучающих текстах, будет называться

T-ошибкой. Точнее говоря, в экспериментах фиксировались не ошибки, а эффективность классификации (см. предыдущий раздел). Таким образом, результат тестирования каждого классификатора на данном узле дерева каталога выражался в виде трех значений эффективности T, K и U.

Эффект от применения каждого из изучаемых методов оценивался в виде двух параметров. Первый из них выражает величину этого влияния и определяется как разность между эффективностью классификации при условии применения этого метода и средней эффективностью, достигнутой альтернативными (т.е., лежащими на той же оси пространства оптимизации) методами. При этом эта величина усреднялась по 8 экспериментам на разных узлах одного и того же уровня каталога. Второй параметр выражает статистическую значимость положительного или отрицательного влияния исследуемого метода. Он основан на регистрации количества раз из числа этих 8 экспериментов, когда данный метод показывает наилучшую или наихудшую эффективность среди альтернативных методов. При этом проверялась гипотеза о равновероятности показа альтернативными методами наилучшего (или наихудшего) значения эффективности. Задавшись уровнем достоверности 0.03, нетрудно вычислить, что в случае бинарной альтернативы (методы **STOP, NOUN, FREQ, TFIDF**) эффект от применения метода должен считаться значимым, только если он проявляется во всех 8 экспериментах. В случае 3 альтернативных методов **WF**/**NF**/**SYN** влияние метода считается значимым, если этот метод является наилучшим или наихудшим не менее, чем в 7 экспериментах из 8. Для различных алгоритмов классификации, где число альтернатив равно 6, критическое число экспериментов с наилучшим или наихудшим исходом равно 5.

Эффект применения различных методов классификации в терминах этих двух параметров представлен в Таблицах 2 и 3. Если применение метода чаще приводило к получению максимальной эффективности, это индицируется положительным значением соответствующего числа в Таблице 3; в противном случае это значение отрицательно. Если максимальная и минимальная эффективность достигалась одинаково часто, это показывается значением в таблице, равным 0. Статистически значимые результаты обозначены выделением соответствующих клеток таблиц и жирным шрифтом.

Первый общий вывод, который можно сделать на основании этих таблиц, подтверждает мою гипотезу о том, что задачи классификации, соответствующие трем уровням каталога, существенно

Таблица 2. Эффект применения отдельных методов на эффективность классификации для 3 уровней каталога.

Уровень	1			2			3		
Тип ошибки	T	K	U	T	K	U	T	K	U
WF	-8.6	0.3	0.5	**27.0**	**24.5**	-6.9	3.7	-2.7	-4.2
NF	0.2	0.4	-0.1	-11.4	-8.1	3.5	-2.6	**-26.4**	-1.5
SYN	8.4	-0.7	-0.5	-15.6	**-16.5**	3.4	-1.1	**29.0**	5.7
STOP	1.6	0.7	0.1	-2.8	-5.6	**-8.4**	**-4.2**	0.1	**4.3**
FREQ	-0.3	-0.5	-0.2	-1.7	4.0	-13.3	-1.8	-2.6	3.8
TFIDF	-0.5	-0.7	0.8	-1.3	0.8	-6.1	-0.7	0.3	2.6
NOUN	-0.5	0.4	0.7	0.5	-0.9	-5.3	0.6	-0.5	5.0
CLLR	-8.8	-11.6	-3.4	-3.8	-6.3	0.0	-2.5	0.5	-6.2
DT	3.7	-8.0	**-13.6**	-2.1	-5.7	-12.8	-8.4	6.9	-6.2
NB	-10.3	-1.7	5.3	-16.8	-3.9	-9.3	-3.6	**-40.5**	8.4
NN	-5.0	-2.6	2.0	-9.4	4.2	-8.8	-4.5	**6.8**	-1.4
RF	8.0	11.7	0.4	8.2	**12.3**	**34.1**	-9.5	6.9	16.5
SVM	**12.4**	12.1	**9.3**	**23.9**	-0.7	-3.2	**28.5**	19.5	-11.1

Таблица 3. Количество экспериментов, показавших при применении данного метода наилучшую или наихудшую эффективность классификации, для 3 уровней каталога.

Уровень	1			2			3		
Тип ошибки	T	K	U	T	K	U	T	K	U
WF	-5	3	4	**8**	**8**	-4	4	-1	-2
NF	2	5	-2	-4	0	5	-1	**-7**	-2
SYN	5	-4	-5	-4	-8	3	-4	**8**	3
STOP	4	4	4	-5	-7	**-8**	**-8**	4	**8**
FREQ	-6	-5	-5	-4	4	-6	-6	-7	6
TFIDF	-6	-5	4	-7	4	-5	-7	5	6
NOUN	-5	5	7	5	-4	-7	6	-5	7
CLLR	-3	-3	0	-1	-4	1	2	0	-1
DT	1	-2	**-8**	1	-2	-4	-1	1	0
NB	-3	0	0	-3	0	-1	0	**-8**	2
NN	-2	1	0	-2	1	-1	-1	**6**	1
RF	1	2	0	2	**7**	**7**	-4	0	3
SVM	**5**	2	**8**	**5**	-1	-1	**6**	1	-3

различаются и по силе влияния на эффективность классификации вариации применяющихся методов, и по наборам оптимальных методов, и по соотношению K- и U-ошибок. В этом нет ничего удивительного, если учесть, что на первом уровне речь идет о классификации нескольких десятков тысяч документов абсолютно разного содержания, в то время как на третьем уровне это классификация сотен или даже только десятков тематически однородных документов. Второй уровень занимает в этом смысле промежуточное положение. Видно, например, что в отличие от остальных двух уровней, результаты классификации на первом уровне вообще малочувствительны к выбору пространства признаков. Еще одно отличие первого уровня – это высокая корреляция K- и U-ошибок.

Вообще, анализ K- и U-ошибок подтверждает мой тезис о необходимости их отдельного рассмотрения. В то время как K- и U-эффективность на первом уровне и K-эффективность на втором уровне, хотя и будучи весьма неоднородной, была как правило высока и редко опускалась ниже 50%, U-эффективность на втором уровне была ниже на 20-30%, а U-эффективность на третьем уровне и вовсе часто была отрицательной. С этим согласуются и данные обеих таблиц. Видно, что величины в последнем столбце, соответствующем U-эффективности на третьем уровне, высоко хаотичны и определяются, скорее всего, случайными факторами. Эти результаты, кстати говоря, приводят к пессимистическому выводу касательно перспектив автоматической классификации документов с новых хостов в узлы каталога третьего уровня.

Перейдем к более подробному анализу Таблиц 2 и 3.

1. Во-первых, отметим, что методы **NOUN, FREQ** и **TFIDF** вообще не оказывают значимого влияния на точность классификации. Это неожиданный результат, по крайней мере, для метода **TFIDF**, который сейчас является общепринятым. Таким образом, для целей автоматического пополнения веб-каталогов, по-видимому, всегда достаточно формировать векторы признаков из нулей и единиц, индицирующих отсутствие или наличие данного объекта в тексте без учета его частоты. Обнаруженная малая эффективность методов **NOUN** и **FREQ** также имеет практические следствия, так как эти методы базируются на лингвистических ресурсах и методах, которые могут быть доступны не для всех интересующих языков.

2. Как уже говорилось, для классификации текстов в узлы первого уровня способ перевода текстов в вектор признаков не важен (разумеется, с точки зрения исследованных здесь методов).

3. При решении задач классификации на втором уровне не нужно делать морфологический анализ и использовать стоп-лист, а

учитывать все словоформы всех слов. Сразу оговорюсь, что это, может быть, единственное место данного исследования, где перенос результатов, полученных на англоязычных текстах, на русскоязычные тексты вероятно неправомерен. Это связано с тем, что русский и английский языки кардинально отличаются по количеству разных словоформ, соответствующих одной нормальной форме, - в русском языке словоформ на порядок больше. Решение вопроса о переносимости этого результата на русскоязычные тексты – это одна из тем дальнейших исследований.

4. На третьем уровне, вследствие малости обучающих выборок и наличия избыточных размерностей пространства признаков, методы понижения этой размерности начинают играть свою роль. По этой причине рекомендуется использовать стоп-лист и тезаурус (методы **STOP** и **SYN**).

5. Метод **SVM** является абсолютным чемпионом по точности описания данных, составляющих обучающий набор (Т-эффективность). Однако, он является лучшим методом для классификации новых текстов только для уровня 1.

6. Неожиданностью оказалась высокая эффективность классификатора на основе случайного леса (**RF**) на уровнях 2 и 3 – как раз в тех случаях, где высока избыточность размерностей векторов признаков. Данный метод до сих пор редко использовался для классификации текстов. Возможно, столь высокая его точность связана с примененной нами его модификацией - заменой критерия Джини на критерий взаимной информации, однако точный ответ на этот вопрос требует дополнительных исследований.

7. В задачах классификации третьего уровня наиболее устойчивые результаты давал алгоритм ближайших соседей (**NN**),

8. Плохие (или неустойчивые) результаты показали методы **CLLR, DT** и **NB.**

В заключение подчеркнем, что в настоящей работе все эти методы оценивались только с точки зрения их точности. За бортом рассмотрения остались такие их аспекты, как время построения классификатора и его применения для классификации нового текста, требования этих методов к объему памяти и пр. В оправдание нужно сказать, что учет еще и этих параметров в нашей оптимизационной задаче сделало бы вероятно невозможным ее решение в разумные сроки. С другой стороны, важность этих аспектов эффективности методов несколько уменьшается, если учесть все время растущую производительность компьютеров, возможность эффективного применения стратегий сэмплинга данных при решении задач первого уровня, а также естественную параллелизуемость зада-

чи построения глобального классификатора для пополнения веб-каталога вследствие ее разбиения на независимые задачи бинарной классификации, которые могут быть, например, разнесены по десяткам компьютеров в локальной сети. Именно такой подход, кстати говоря, и был использован мной в данном исследовании.

5. Заключение и практические рекомендации по реализации процедуры автоматического пополнения веб-каталога

Итак, мной было исследовано влияние применения разных методов на точность решения задач классификации текстов, возникающих в процессе решения проблемы автоматического пополнения веб-каталогов. При этом нашли подтверждение два следующих предположения методического характера:

- на разных уровнях иерархии каталога наиболее эффективными оказываются разные наборы методов классификации;
- для адекватной оценки точности классификации надо использовать для тестирования классификаторов не только документы хостов, содержащих обучающие документы (оценка K-ошибки), но и документы с хостов, не содержащих обучающие документы (оценка U-ошибки), так как эти оценки могут очень сильно различаться особенно для нижних уровней дерева каталога.

Что касается рекомендаций по выбору методов классификации, то в результате проведенного исследования можно прийти к следующим выводам.

- **на всех уровнях**

  - Рекомендуется использовать простой бинарный метод формирования векторов признаков без учета частот, при котором наличие какого-то объекта в тексте обозначается единицей в соответствующей позиции вектора признаков, а отсутствие обозначается нулем.

- **на уровне 1**

  - Нет необходимости проводить морфологический анализ или пользоваться тезаурусами или какими-либо методами

уменьшения размерности пространства признаков для повышения точности классификации, однако можно применять все эти методы, если это необходимо по соображениям экономии памяти.

- Классификаторы лучше строить с помощью метода **SVM**, а при жестких ограничениях на времена счета – существенно более быстрым методом **NB**.

- на уровне 2

- Не рекомендуется применять какие-либо методы понижения размерности пространства признаков.

- Рекомендуется применять метод **RF** для построения классификаторов.

- на уровне 3

- Рекомендуется применять индикаторы наличия или отсутствия слов – представителей тех или иных синонимических групп в тексте в качестве элементов вектора признаков (используя тезаурус), а также использовать стоп-лист.

- Рекомендуется применение методов **NN** (с оптимизацией параметров с помощью генетического алгоритма) или **RF**.

Настоящие рекомендации были получены на основе экспериментов с англоязычным подмножеством наполнения каталога Яндекс. В дальнейших исследованиях, проводимых по мере реализации все новых и новых методов классификации для языков, отличных от английского, мы собираемся проверить зависимость этих рекомендаций от языка классифицируемых документов.

Литература.

[1] Breiman, L. ‘Random Forest’, Machine Learning, 45, 2001, pp 5-32
[2] Cover, T. and Thomas, J. ‘*Elements of Information Theory*’, Wiley, 1991

[3] Cristianini, N., Shawe-Taylor, J. 'An Introduction to Support Vector Machines (and other kernel-based learning methods)', Cambridge University Press, 2000
[4] Dhillon, I., Mallela, S., Kumar, R. 'Enhansed Word Clustering for Hierarchical Text Classification', Proceedings of KDD2002, 2002, pp 191-200
[5] Dumais, S. and Chen, H. 'Hierarchical Classification of Web Content', Proceedings of 23rd ACM Int. Conf. RDIR, 2000, pp 256-263
[6] Fellbaum, C. (editor), '*WordNet: An Electronic Lexical Database*', MIT Press, 2005
[7] Goldberg, D. '*Genetic Algorithms*', Addison Wesley, 1988
[8] Greene, W. '*Econometric Analysis*', Prentice Hall, 1997
[9] Mitchell, T. '*Machine Learning*', McGraw Hill, 1997
[10] PolyAnalyst data/text mining system. User manual. http://www.megaputer.com
[11] Zhang, D. and Lee, W. 'Web Taxonomy Integration using Support Vector Machines', Proceedings of WWW2004, 2004, pp 472-481

[1] www.yandex.ru